



King's Research Portal

DOI:

[10.1109/NETSOFT.2016.7502440](https://doi.org/10.1109/NETSOFT.2016.7502440)

Document Version

Peer reviewed version

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Mapp, G., Sardis, F., & Crowcroft, J. (2016). Developing an implementation framework for the Future Internet using the Y-Comm architecture, SDN and NFV. *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 2016(June), 43-47. [7502440]. <https://doi.org/10.1109/NETSOFT.2016.7502440>

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Developing an Implementation Framework for the Future Internet using Software Defined Networking and the Y-Comm Architecture

Glenford Mapp
School of Science
and Technology
Middlesex University,
Hendon, London NW4 4BT
Email: g.mapp@mdx.ac.uk

Fragkiskos Sardis
Department of Informatics
Kings College London
London, WC2R 2LS, UK
Email: fragkiskos.sardis@kcl.ac.uk

Jon Crowcroft
Computer Laboratory
William Gates Building
JJ-Thomson Avenue
Cambridge, UK
Email: jon.crowcroft@cl.cam.ac.uk

Abstract—The Future Internet will provide seamless connectivity via heterogeneous networks. The Y-Comm Architecture is a reference model that has been developed to build future mobile systems for heterogeneous environments. However, the emergence of Software Defined Networking and Network Functional Virtualization will allow the implementation of advanced mobile architectures such as Y-Comm to be prototyped and explored in more detail. This paper proposes an implementation model for the Y-Comm architecture based on these mechanisms. A key component is the design of the Core Endpoint which connects various peripheral wireless networks to the core network. This paper also proposes the development of a Network Management Control Program which allows the management routines running in the Cloud to control the underlying networking infrastructure. The system being proposed is flexible and modular and will allow current and future wireless technologies to be seamlessly integrated into the overall system.

Keywords—Future Internet, Software Defined Networking, Y-Comm Framework, Quality-of-Service, Cloud Computing, Core and Access Networks

I. INTRODUCTION

The Internet of the Future will place new demands on common networks. There will be support for ubiquitous communication via heterogeneous networking. Users will expect networks to support access to data using the 4As paradigm: anytime, anywhere, anything, anyhow. In order to facilitate these demands, network evolution is taking place which divides the Future Internet into core and peripheral networks. These two components are taking two different evolutionary paths: the core of the network will be dominated by the deployment of fast optical networks while peripheral networks will support a number of different access technologies including WiFi, LTE, 5G and WiMax. There is an increasing acknowledgement of the need of the mobile industry to get to grips with these developments [1].

In addition, the need to support a diverse set of applications with different Quality-of-Service (QoS) requirements means that support for QoS needs to be incorporated into future networks. In this new environment, QoS will also include security requirements and the system should be able to balance between the security and QoS concerns.

Y-Comm [2] is an architecture that has been designed to build heterogeneous mobile networks. It attempts to integrate communications, mobility, QoS and security into a single platform. It divides the Future Internet into two frameworks: the Core Framework and the Peripheral Framework. The researchers of Y-Comm have made major contributions in the areas of proactive handover [3] as well as introducing new concepts in security such as Targeted Security Models [4]. Other network architectures for mobile systems such as Hokey [5] and Ambient Networks [6] have also been explored but Y-Comm offers the most functionality and flexibility [7] while integrating various key mechanisms [8], [9].

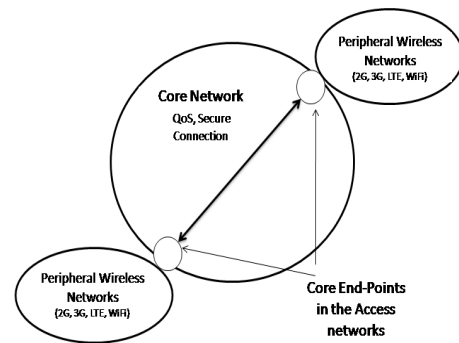


Fig. 1: Internet Evolution

However, the emergence of Software Defined Networking (SDN) [10] means that it is now possible to build mechanisms that can implement the Core Framework in Y-Comm leading to a workable prototype of the Y-Comm architecture. This paper contributes towards this goal by first exploring the design of an Implementation Framework for Y-Comm, it then defines the operational structure of the Core Endpoint, a key part of the Y-Comm design. Finally, the functionality of a

Network Management Control Protocol (NMCP) which links the network management routines running in the Cloud to the underlying networking infrastructure at the Core Endpoint is discussed.

The rest of the paper is structured as follows: Section 2 explores the concept of network evolution leading to the Y-Comm Reference Model. Section 3 investigates the Core Evolution in more detail, while Section 4 presents the Implementation model for Y-Comm. Section 5 details the operational structure of a Core Endpoint. Section 6 looks at the Network Management Control Protocol (NMCP) while Section 7 discusses the current work being done on the project. The paper concludes in Section 8.

II. NETWORK EVOLUTION

The Internet is currently evolving. Instead of large, global, but individually-managed networks, a core network is being deployed which is fast and getting faster with peripheral wireless networks situated at its edges. A Core Endpoint is an entity which is at the edge of the core network and is used to connect different types of wireless systems. The set-up is shown in Figure 1. A mobile node, which is now usually a smart phone, can have several interfaces including WiFi, LTE and WiMax. Vertical handover techniques between different interfaces are used to maintain the connection between the mobile node and the core network via the Core Endpoint as the user moves around.

A. The Y-Comm Architecture: The Reference Model

The Y-Comm architecture is an architecture for mobile computing. The architecture is shown in Figure 2 and is specified in [2]. Below a brief summary is given, starting with the lower layers:

- **Layer 1: The Hardware Platform layer (HPL):** This layer is used to define the hardware components and technologies required to support a particular wireless network, including the electromagnetic spectrum, modulation techniques, Media Access Control (MAC) algorithms.
- **Layer 2: The Network Abstraction layer (NAL):** This layer specifies a common networking interface which all networks employing this architecture must support. This layer is used to maintain and control the network interfaces on the mobile node.
- **Layer 3: The Handover Management layer (HML):** This layer implements the vertical handover mechanism. So this layer acquires the resources for handover, does the handover signalling, context transfer and the initial packet reception after vertical handover.
- **Layer 4: The Mobility Management layer (MML):** This layer is used to monitor the mobility of the mobile and evaluates all the circumstances to determine when handover should occur.
- **Layer 5: The End Transport System (ETS) :** This layer looks at moving data to and from the mobile node. Since most peripheral networks will be wireless, it is therefore important to ensure that network and

transport systems operate efficiently so that applications running on the mobile node can receive a sustainable quality-of-service.

- **Layer 6: The QoS Layer (QoSL):** This layer helps to ensure that the Quality-of-Service required by applications can be maintained as the quality-of-service being offered by the networks is dynamically changing as the mobile node moves around.
- **Layer 7: The Application Environments layer (AEL):** This layer specifies mechanisms and routines that allow applications to be built which can use all the layers of the framework.

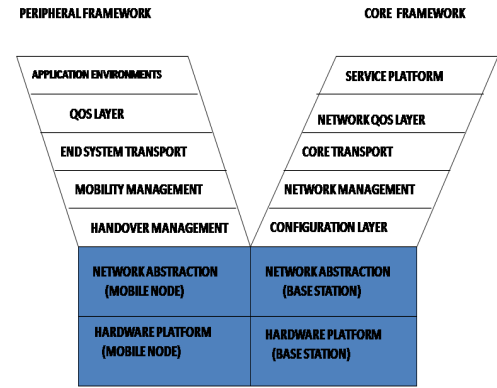


Fig. 2: The Y-Comm Framework: The Reference Model

B. The Core Framework

In this section, the Core Framework is discussed. The first two levels of the Core Framework are similar in purpose to the first two layers of the Peripheral Framework. However, while the Peripheral Framework specifies software such as device drivers needed to support a given network on a mobile node, in the Core Framework these layers represent the functionality that software modules need in order to manage network resources, such as radio channels on base-stations of a given technology.

- **Layer 3: The Configurable Layer (CL):** This layer makes use of programmable networking techniques to control key infrastructural hardware, including routers and switches in the core network as well as access routers and base-stations.
- **Layer 4: The Network Management Layer (NML):** This layer acts as a management plane that uses the Configuration Layer to bring together various hardware and software components to build enterprise class networks. The NML also stores information on the peripheral networks in the local area.

- **Layer 5: The Core Transport System (CTS):** This layer is about network addressing and transport mechanisms in the core network. The Core Transport system is mainly concerned with moving data having a given QoS and security requirement between two core endpoints in the core network.
- **Layer 6: The Network QoS layer (NQL):** This layer is responsible for QoS issues within the core network. It looks at how QoS may be defined and the mechanisms used to establish and maintain QoS at different points in the system [11].
- **Layer 7: The Service Platform layer (SPL):** The Service Platform layer allows different agents to install and operate various services in a secure and controlled fashion. The service platform will provide support for services which have mobile clients.

III. CORE EVOLUTION

Previous work on developing key mechanisms for Y-Comm was focused on the Peripheral Framework. This has included work on handover classification, proactive mobility mechanisms, transport protocols, Quality-of-Service (QoS) and security. However, recently there have been developments in the Core Framework.

A. New Hybrid QoS Framework

Work has looked at developing a new Hybrid Quality-of-Service, based on the integration of the ideas of IntServ [12] and DiffServ QoS Models [13], [14]. The new network QoS model [15] uses individual flow management via IntServ between the Mobile Node and the Core Endpoint but uses traffic classes based on DiffServ to move the data between Core-Endpoints. The mapping between individual flows and traffic classes is done by the QoS Broker running in the Core Endpoints. Figure 3 shows how that mapping is done during connection setup. IntServ flows are denoted by flow_ids on the the client and server networks which are mapped to DiffServ Classes at the respective Core Endpoints in the core network.

B. A Service-Oriented Framework for Mobile Services

The second development is a new Service Architecture that allows the development of mobile services using the Cloud [16]. This means that services can be managed, copied or migrated to support mobile users. The system provides algorithms that incorporate traffic management as well as the QoS requirements of the flow. This new Framework [17] has six layers which, from the top to the bottom, are briefly described below:

- **The Service Management Layer:** This layer manages the service that is being provided. It specifies the functions of the service, registers the service in a service registry and obtains a unique service_id. It also specifies the minimum resources required by networking and cloud infrastructure needed to run the service in terms of computing resources, network Quality-of-Service requirements and storage needs.
- **The Service Subscription Layer:** This layer handles the functions required for global clients to use the

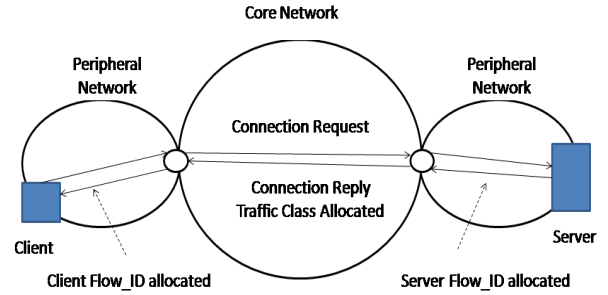


Fig. 3: Connection Setup using the New QoS Model

service. It therefore allows clients to subscribe to services. It provides the user with a unique client_id, a given Service Level Agreement (SLA) and sets up accounting and payment mechanisms.

- **The Service Delivery Layer:** The layer is in charge of delivering the service to a given client. It first maps the SLA to a given QoS and ensures that the selected server as well as its networks can meet the required QoS. The service also receives notifications and triggers about handovers and based on these notifications, this layer may replicate or migrate the service closer to the user.
- **The Service Migration Layer:** The layer handles the replication or migration of services to different cloud platforms to facilitate a good Quality-of-Experience for the mobile user. Migration is done at the request of the Service Delivery Layer.
- **The Service Connection Layer:** This layer is responsible for managing the connection between a client and the service and reports changes in transport or network parameters such as bandwidth or delay to the Service Delivery Layer.
- **The Network Abstraction Layer:** This layer allows the service to interface to different types of networks, depending on network architecture and addressing. In the current Internet, this layer maps onto IP networking with TCP/IP. In more advanced systems such as Y-Comm, this functionality is spread between the QoS and Transport Layers in Core and Peripheral Frameworks.

Though these developments are significant, however, what is missing is a way of prototyping the layers of the Core Framework.

C. Using Software Defined Networks

Various papers have shown how the Core Framework can be mapped onto current core network technologies such as 3G [18], UMTS [19] and LTE [20]. However, new technologies such as SDN, NFV and Cloud systems make it possible to implement the layers of the Core Framework. Thus the use of these mechanisms will allow Y-Comm to be implemented as an example of Network-as-a-Service (NaaS) [21]. The Network Abstraction layer can be implemented using the 802.21 protocol [22] in the Access Routers and Base Stations; this will allow channels to be allocated to mobile nodes as users move around. The Configuration Layer which would comprise the current ISP networks is being replaced by networks that support OpenFlow which could be used to dynamically connect flows. The Network Management layer is implemented using a Cloud System to run the required functions of that layer.

IV. AN IMPLEMENTATION MODEL FOR Y-COMM

These changes have allowed us to develop an Implementation Model as shown in Figure 4. In Peripheral Framework, the key mechanisms include:

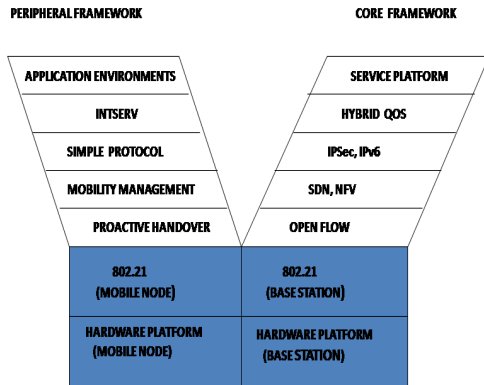


Fig. 4: The New Implementation Model for YComm

The Peripheral Implementation Model include:

- **IEEE 802.21 mechanism:** This layer, Layer 2, uses the IEEE 802.21 mechanisms [23] to control the different wireless interfaces. Higher layers can send commands to bring up or bring down links and network interfaces signal events via L2 triggers about the availability of individual networks.
- **Proactive Handover:** This attempts to do proactive handovers using Time Before Vertical Handover (TBVH) and Network Dwell Time (NDT) parameters based on data from the mobility management layer.

- **Mobility Management:** There are several mechanisms that can be used for monitoring the mobility of mobile nodes. This is based on triggers from network interfaces as well as GPS readings and parameters.
- **Simple Protocol:** The Simple Protocol (SP) [24] is a new transport protocol between the Mobile Node and the Base Station. SP also uses ECHO packets to determine the bandwidth and latency of individual connections. This allows applications on the mobile node to react to changes in the Quality-of-Service due to core network changes or vertical handovers. A secure version of SP has also been investigated [25]. In addition, the use of a Location/ID based addressing scheme should also be used to allow efficient communication using different network interfaces in a heterogeneous environment [26],[27].
- **IntServ:** This is a version of IntServ which allows the mobile node to specify its QoS requirements on a per-flow basis.
- **The Application environment:** This allows applications to use an enhanced IntServ specification to specify its Quality-of-Service requirements on individual connections. This layer therefore takes application requirements expressed as Service Level Agreements (SLAs) and turns them into QoS requirements for both the application and the network.

The Core Implementation Model include:

- **Access Point/Base Station Controllers running 802.21:** This allows the Base Station and Access Routers to be controlled using an enhanced IEEE 802.21 protocol.
- **OpenFlow:** This is used for reconfigure the access network routers depending on the mobility of the user.
- **Software Defined Networking (SDN) and Network Functional Virtualization (NFV):** This is used to implement the mechanisms and services for the Network Management Layer and is explored in detail below.
- **IPSec, IPv6:** The TCP/IP Suite is maintained but IPSec [28] is used to ensure that secure tunnels can be set up between Core Endpoints.
- **Hybrid QoS:** This is used to integrate IntServ and DiffServ mechanisms. The mapping between IntServ and DiffServ parameters is done at Core Endpoints.
- **Service Platform Layer:** This implements the Mobile Services platform described above.

V. BUILDING A CORE ENDPOINT

The key to realizing a prototype Y-Comm testbed is the design and development of a real Core Endpoint based on the Implementation model described above. The management functions run in the Cloud. The operational structures of the Core Endpoint are shown in Figure 5 and the functions are detailed below:

- **Core Routers:** These routers attach the Core Endpoint to the core network. They are used to route data between the core and peripheral networks.

- **Routing Engine:** This is used to manage routes between the Core Endpoints and mobile nodes as well as routes between Core Endpoints.
- **Network Topology Manager:** This service maintains the topology of the network in the local area. This service handles all changes to network infrastructure including changes of points of attachment of mobile nodes due to handover, the installation of new routers and switches as well as link and network failures.
- **Event and Trigger Manager:** This service deals with events and triggers that happen at the lower layers of the architecture.
- **QoS Broker:** The QoS Broker is used to manage the QoS issues for Core Endpoints. This service ensures that the incoming or outgoing flows do not exceed the network capacity of the Core-Endpoint and does the mapping between IntServ flows and DiffServ traffic classes.
- **The AAAC Service:** This manages the security of the Core Endpoint: This includes all aspects to do with the use of the networking infrastructure.
- **Third Party Services:** These services allow clients of mobile operators to use the local Cloud infrastructure to offer bespoke services to their clients. This may be extra computing or storage facilities. New Services such as caching of data content in local clouds for Context Data Networks (CDN) could also be installed.

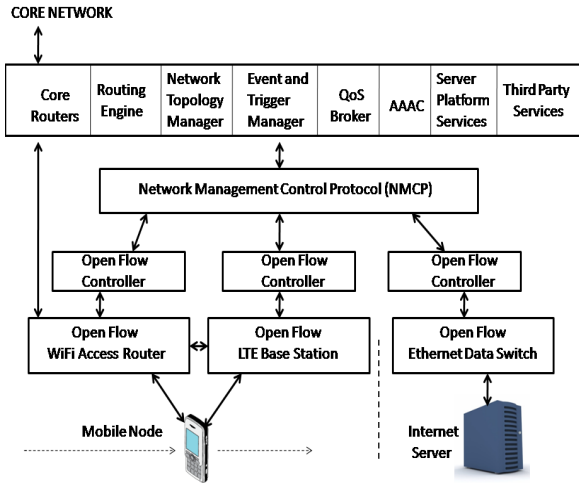


Fig. 5: The Operational Structure of a Core Endpoint

These management routines run in a Cloud environment. They communicate with each other the NETCONF protocol [29] running over secure SP. This approach greatly enhances the evolution of the network as individual services can be updated or changed without affecting the other management services. For example, it should be possible to change the routing engine to do Information Centric Networking (ICN) routing without having to change other services in the Network Management Layer.

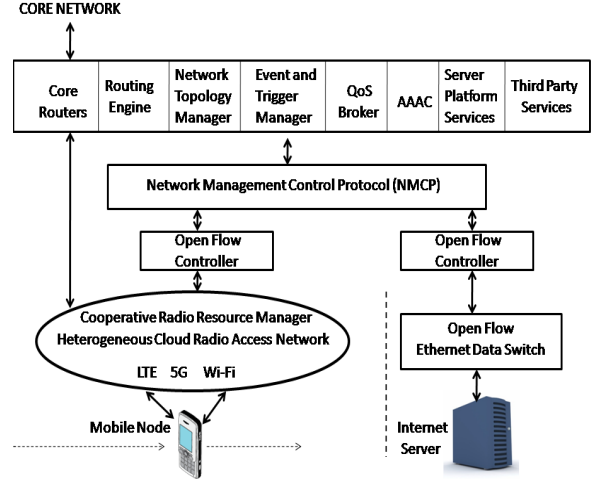


Fig. 6: Showing the Evolution of Networking Technologies

A. Coping with the Evolution in Mobile Technologies

In this section, we show how the proposed system will evolve with the development of new mobile technologies. This evolution will allow the smooth management of local heterogeneous networks by the Heterogeneous Cloud Radio Access Network (H-CRAN) and the Cooperative Radio Resource Manager (CRRM). H-CRAN will be used to access and control individual networks which CRRM will be used to optimize the overall radio access environment. CRRM will also support OpenFlow and hence the upper layers of the architecture can remain unchanged. The new structure is shown in Figure 6.

VI. THE NETWORK MANAGEMENT CONTROL PROTOCOL

The Network Management Control Protocol is used to allow the high-level network management functions and services discussed above to control and manage networking infrastructure directly. The protocol therefore acts on all the entities of the network infrastructure including: Core Routers, Ethernet Controllers which in turn control Ethernet Data Switches, Base-Station and Access Controllers.

NMCP also supports various communication entities: An Endpoint is a device that can send or receive data. Mobile nodes and servers are examples of endpoints. A link is a connection between two entities; it could be a wired or wireless link. A path is a connection between two endpoints. A path is made up of one or more links. A data-flow is the movement of data between two endpoints. A data-flow may be unipath; i.e., there is one path between two endpoints; while a multipath flow indicates that there are several paths between two endpoints. In addition, flows can be divided into two types: a single flow or a compound flow which is made up of more than one flow. Data-flows allow us to specify the actual data flowing along physical links. Making these flows first-class objects allows for much greater flexibility in how the network is managed. NMCP is therefore concerned with specifying end-to-end connections, paths and data-flows. NMCP supports a number of commands which are sent to the Open Controller

using systems such as Floodlight which in turn translates these commands to Open Flow messages to the Access Routers and Ethernet Data switches. NMCP commands include:

- **Attach-Link:** This function is used to create a link between two entities.
- **Detach-Link:** Removes a link between two entities
- **Act-Link:** Activates a link to forward packets
- **Deact-Link:** Deactivates the link
- **Create-Path:-** Creates a path between two endpoints
- **Modify-Path:-** Modifies a path between two endpoints
- **Delete-Path:-** Removes a path between two endpoints
- **Create-Flow:-** Creates a data-flow between two endpoints
- **Modify-Flow:-** Modifies the characteristics a data-flow
- **Delete-Flow:-** Deletes a data-flow
- **Merge-Flow:-** Merge a data-flow into another (compound) data-flow
- **Demerge-Flow:-** Separate a data-flow from a compound data-flow
- **Get-parameters:-** Obtain parameters on a link, flow, router or switch
- **Set-parameters:-** Set parameters or thresholds on a link, flow, router or switch
- **Set-Note-Event:-** Notify the management routines about an event on a link, flow, router, switch or network.
- **Delete-Note-Event:-** Delete an event notification on a link, flow, router, switch or network.

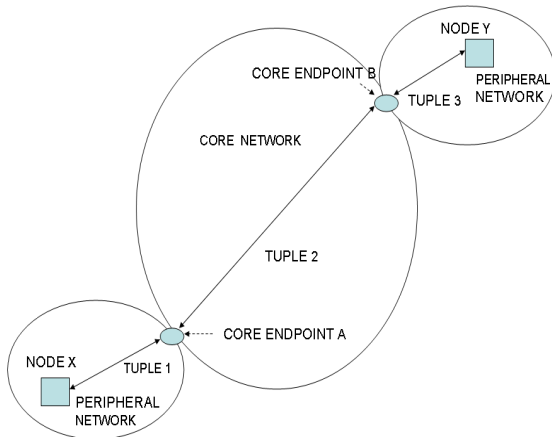


Fig. 7: Connection Setup using NMCP

VII. CONNECTION SETUP USING NMCP

In this section we show an example of using NMCP. The connection setup is shown in Figure 7. A link in NMCP is represented by a tuple containing the entities that are linked together and a label referred to as a **forward connection label** or fcl. The fcl contains details of the physical characteristics such as speed of the link as well as QoS and security information related to the connection. Fcls are therefore treated as capabilities and cannot be tampered with. So in order for packets to be forwarded on a link, it must have a valid fcl. So let us represent this interaction in the following tuple format: **src, fcl, destination**.

We now show how data will be routed from Node X to Node Y, via the core endpoints Core A and Core B. A connection request is first sent by X to Core-Endpoint A; then the related tuples for the connection are created by the Attach-Link command:

Connection Request(Node X,Node Y)

TUPLE 1=Attach-Link(Node X,fcl(XA),Core A)

This allows Node X to send packets to the Core Endpoint A.

TUPLE 2=Attach-Link(Core A,fcl(AB), Core B)

This tuple allows Core endpoint A to send data to Core Endpoint B.

TUPLE 3=Attach-Link(Core B,fcl(BY), Node Y)

The final tuple allows Core Endpoint B to send packets to Node Y.

So we can now create a path between X and Y by joining the link tuples:

Path(XY)=Create-Path(TUPLE 1 + TUPLE 2 + TUPLE 3)

We can now create a data-flow using this path:

Data-flow(XY)=Create-Flow(Path(XY),XA, Merge-Flow(XA,AB),BY = Demerge-Flow(AB,XA))

This means that the data-flow moves data along the physical path denoted by PATH(XY). However, the specification indicates that the initial flow XA can be merged into a compound flow that is going from Core Endpoint A to Core Endpoint B. The flow is then de-merged at Core Endpoint B and forwarded to Node Y.

Once the data-flow has been created; the connection is started by activating all the relevant links if they are not already activated.

Conn(XY)=Act-Link(TUPLE 1 + TUPLE 2 + TUPLE 3)

Finally, a message is sent to X to indicate that the Connection has been created.

fcl(X,A)= Connection_granted(Node X,Node Y)

So fcl(X,A) is now used by Node X to send data to Node Y using the path that has been set up for that connection.

A. Handling Handover

In this section, we show how handover is handled by NMCP. The setup is shown in Figure 8. Node X decides to handover from Core Endpoint A to Core Endpoint C. It sends a handover request to Core Endpoint A.

Handover_request(X,Core A: X,Core C)

This means that Node X wants to be handed over from Core Endpoint A to Core Endpoint C. Hence the Connection(XY) needs to be re-routed. Core Endpoint A sends a request to Core Endpoint C to create two new links:

NewTUPLE 1=Attach-Link(Node X,fcl(X,C),Core C)

NewTUPLE 2=Attach-Link(Core C,fcl(C,B),Core B)

We now need to modify the path from X to Y:

**NewPath(XY)= Modify-Path(XY,NewTUPLE 1
+ NewTUPLE 2 + TUPLE 3)**

**NewData-flow(XY)= Modify-Flow(NewPath(XY),XC,
Merge-Flow(XC,CB),BY= Demerge-Flow(CB, XC))**

We can now activate the new links:

NewConn(XY)=Act-Link(NewTUPLE 1,NewTUPLE 2)

In order to deallocate the old data-flow and the related links we ask that a notification be generated when a new packet is received on the new link:

Set-Note-Event(PACKET_ARRIVAL,NewTUPLE 1)

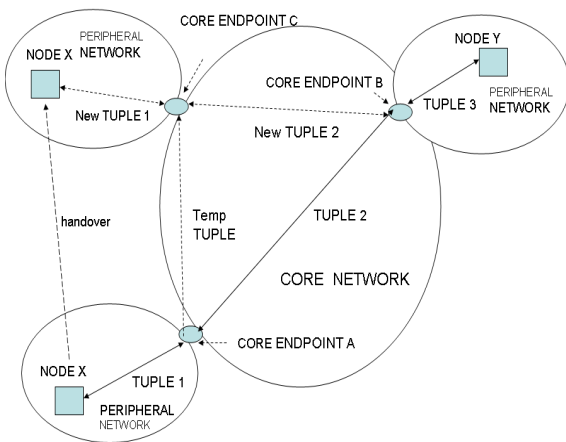


Fig. 8: Core Endpoint Handover Using NMCP

We then set a Conditional Execute instruction to handle this event:

Conditional Execute:

(Event == PACKET_ARRIVAL,NewTUPLE 1)

begin action 1

Delete-Path(XY)

Delete-Flow(XY)

Path(XY) = New Path(XY)

Data-flow(XY) = NewData-flow(XY)

Conn(XY) = NewConn(XY)

Deact-link(TUPLE 1)

Delete-Link(TUPLE 1)

end action

However, before handover starts we must specify that packets sent from Node Y to Node X during handover can no longer be routed to Node X using TUPLE 1 as the node is no longer in that network. So packets reaching Core Endpoint A must be routed to Core Endpoint C. So we must first create a temporary Link between Core Endpoint A and Core Endpoint C.

TempTUPLE= Attach-Link(Core A,fcl(A,C),Core C)

**TempPath(YX)= Modify-Path((YX),TUPLE 3 + TUPLE 2
+ TempTUPLE + NewTUPLE 1)**

**TempData-flow(YX)= Modify-Flow(TempPath(YX),YB,
Merge-Flow(YB,BA),AC= Demerge-Flow(BA,YB),CX)**

So we now activate the new reverse path during handover:

Set-Note-Event (Handover_granted(X,Core A: X,Core C))

Conditional Execute:

(Event == Handover_granted(X,Core A: X,Core C)

begin action

TempConn(YX) = Act-Link(TempTUPLE)

Delete-Path(YX)

Delete-Flow(YX)

Path(YX) = TempPath(YX)

Data-flow(YX)= TempData-flow(YX)

Conn(YX) = TempConn(YX)

end action

We also need to delete this set up when the handover has been completed:

Conditional Execute:

(Event == PACKET_ARRIVAL,NewTUPLE 1)

begin action 2

Delete-Path(YX)

Delete-Flow(YX)

Path(YX) = New Path(XY)

Data-flow(YX) = NewData-flow(XY)

Conn(YX) = NewConn(XY)

Deact-link(TempTUPLE, TUPLE 2)

Delete-Link(TempTUPLE, TUPLE 2)

end action

Finally we send a handover granted reply giving Node X the new fcl to talk to Core Endpoint C:

fcl(X,C)=Handover_granted(X,Core A: X,Core C)

Node X is now able to do the handover. After the handover, it uses fcl(X,C) to send packets to Core Endpoint C and this triggers the conditional execute actions which result in the deallocation of resources.

VIII. CURRENT WORK

Work has begun to specify the first version of the Network Management Control Protocol (NMCP) in terms of structures and message formats. This will then be used to develop a Core Endpoint based along the lines of this paper. The Core Endpoint will use Cloud Facilities via SDN and NFV. It has been decided to initially use the OpenStack Cloud Environment as there has been considerable interest in getting OpenStack to work with OpenFlow as this will allow a prototype version of NMCP and thus a Core Endpoint to be quickly developed. Then more advanced Cloud mechanisms will be explored [30].

IX. CONCLUSIONS

This paper has presented an Implementation Framework for the Future Internet using Software Defined Networking and the Y-Comm Framework. The approach explored will allow the evolution of both underlying network infrastructure in terms of the development of H-CRAN and CRRM as well as the development of new network services by third parties.

REFERENCES

- [1] J. C. Zuniga, C. J. Bernados, M. T. O. A. C. R. and A. Reznik, "Distributed Mobility Management: a Standards Landscape," *IEEE Communications Magazine*, vol. 51, pp. 80–89, March 2013.
- [2] G. Mapp, F. Shaikh, J. Crowcroft, D. Cottingham, and J. Baliosian, "Y-Comm: A Global Architecture for Heterogeneous Networking (Invited Paper)," in *3rd Annual International Wireless Internet Conference (WICON)*, October 2007.
- [3] G. Mapp, F. Katsiriku, M. Aiash, N. Chinnam, R. Lopes, E. Moreira, R. P. Vanni, and M. Augusto, "Exploiting Location and Contextual Information to Develop a Comprehensive Framework for Proactive Handover in Heterogeneous Environments," *Journal of Computer Networks and Communications*, February 2012.
- [4] G. Mapp, M. Aiash, A. Lasebae, and R. Phan, "Security Models for Heterogeneous Networking," in *International Conference on Security and Cryptography (SECRYPT 2010) Athens, Greece*, July 2010.
- [5] HoKey Working Group, "Hokey: Security Support for IEEE 802.21: Media Independent Handover Services," 2010.
- [6] N. Niebert, A. Schieder, A. Abramowicz, H. Malmgen, G. Sachs, J. Horn, C. Prehofer, and H. Karl, "Ambient Networks: An Architecture for Communication Networks Beyond 3G," in *IEEE Wireless Communications*, vol. 11, 2004.
- [7] M. Aiash, G. Mapp, A. Lasebae, J. Loo, F. Sardis, R. Phan, M. Augusto, R. Vanni, and E. Moreira, "A Survey of Potential Architectures for Communication in Heterogeneous Networks," in *Proceedings of Wireless Telecommunications Symposium (WTS 2012)*, April 2012.
- [8] M. Aiash, G. Mapp, A. Lasebae, J. Loo, and R. Phan, "A Formally Verified AKA Protocol For Vertical Handover in Heterogeneous Environments using Casper/FDR," *EURASIP Journal on Wireless Communications and Networking (Open Springer)*, April 2012.
- [9] M. Aiash, G. Mapp, A. Lasebae, R. Phan, and J. Loo, "Integrating Mobility, Quality-of-Service and Security in Future Mobile Networks," *Electrical Engineering and Intelligent Systems: Lecture Notes in Electrical Engineering*, vol. 130, pp. 195–206, 2013.
- [10] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," April 2012.
- [11] A. Duda and C. Sreenan, "Challenges for Quality of Service in Next-Generation Mobile Networks," in *Proceedings of IT&T Annual Conference*, 2003.
- [12] R. Braden, D. Clark, and S. Shenker, "RFC 1633: Integrated Services in the Intranet Architecture: an Overview," in *Technical Report*. IETF, 1994.
- [13] D. Grossman, "RFC 3260: New Terminology and Classification for DiffServ," in *Technical Report*. IETF, 2002.
- [14] J. Crowcroft and Philippe, "Differentiated end-to-end internet services using a weighted proportional fair sharing TCP," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 6, pp. 53–69, July 1998.
- [15] E. Akenunwa, *Improving End Users' Quality of Experience by Ensuring Quality of Service In Heterogeneous Networks*, School of Science and Technology, Middlesex University, October 2013, MSc Thesis.
- [16] Q. Duan, Y. Yan, and A. Vasilakos, "A Survey on Service-Oriented Network Virtualization toward Convergence on Networking and Cloud Computing," *IEEE Transactions on Network and Service Management*, vol. 9, pp. 373–392, 2012.
- [17] F. Sardis, *Exploring Traffic and QoS Management mechanisms to support mobile cloud computing using service localization in heterogeneous environments*, School of Science and Technology, Middlesex University, August 2014, PhD Thesis.
- [18] G. Mapp, F. Shaikh, M. Aiash, R. Vanni, M. Augusto, and E. Moreira, "Exploring Efficient Imperative Handover Mechanisms for Heterogeneous Networks," in *Proceedings of the International Symposium of Emerging Ubiquitous and Persuasive Systems, Indianapolis, USA*, August 2009.
- [19] F. S. Shaikh, *Intelligent Proactive Handover and QoS Management using TBVH in Heterogeneous Networks*, School of Engineering and Information Sciences, Middlesex University, January 2010, PhD Thesis.
- [20] M. Aiash, G. Mapp, A. Lasebae, and A. Al-Nemrat, "Supporting LTE Networks in Heterogeneous Environment using the Y-Comm Framework," in *Proceedings of the 4th International Conference on Networks and Communications*, December 2012.
- [21] Q. Duan, "Network-as-a-Service in Software Defined Networks for end-to-end QoS provisioning," in *Proceeding of the 23rd Conference on Wireless and Optical Communication (WOCC)*. Newark, NJ, USA: IEEE, 2014, pp. 1–5.
- [22] IEEE 802.21 Working Group, "IEEE 802.21: Media Independent Handover Services." [Online]. Available: www.ieee802.org/21/ (Lastaccessed13/09/2015)
- [23] W. Chen, J. Liu, and H. Huang, "Implementation and performance study of IEEE 802.21 in Integrated IEEE 802.11/802.16 e networks," in *Proceedings of the 10th International Conference on Parallel and Distributed Systems*, 2004.
- [24] Riley, L and Mapp, G, "yRFC3: The Specification of SP-Lite." [Online]. Available: http://www.mdx.ac.uk/research/science_technology/informatics/projects/ycmm.aspx (Lastaccessed:30/09/2014)
- [25] K. Kammuehler, G. Mapp, S. Patel, and S. Abubaker, "Engineering Security Protocols with Model Checking - Radius-SHA256 and Secured Simple Protocol," in *Proceedings of the 7th International Conference on Internet Monitoring and Protection*, 2012.
- [26] A. R., "ILNP Concept of Operation," 2008 June, Internet Draft.
- [27] G. Mapp, M. Aiash, H. Guardia, and J. Crowcroft, "Exploring Multihoming Issues in Heterogeneous Networks," in *Proceedings of the First International Workshop on Protocols and Applications with Multihoming Support (PAMS 2011)*, March 2011.
- [28] N. Doraswamy and D. Harkins, "IPSec: The New Security Standard for the Internet, Intranets and Virtual Private Networks," in *Prentice Hall Security Series*. Prentice Hall, 2003.
- [29] R. Enns, M. Bjorkland, J. Schoenwaelder, and A. Bierman, *Network Configuration Protocol*, IETF, June 2011.
- [30] A. Madhavapeddy, R. Mortier, R. Sohan, T. Gazagnaire, S. Hand, T. Deegan, D. McAuley, and J. Crowcroft, "Turning down the LAMP: software specialisation for the cloud," in *Proceedings of the 2nd USENIX conference on Hot Topics in Cloud Computing (HotCloud'10)*. USENIX Association Berkeley, 2010, pp. 11–18.